PERBANDINGAN K-SUPPORT VECTOR NEAREST NEIGHBOR TERHADAP DECISION TREE DAN NAIVE BAYES

Eko Prasetyo [1)] , Rr Ani Dijah Rahajoe [2)] , Arif Arizal [3)]
[1) , 2), 3)] Teknik Informatika Universitas Bhayangkara Surabaya
Jl. A. Yani 114, Surabaya, 60231
E-Mail: eko1979@yahoo.com[1)], anidrahayu@gmail.com[2)], qariff@gmail.com[3)]

Abstrak. Algoritma K-Support Vector Nearest Neighbor (K-SVNN) menjadi salah satu alternatif metode hasil evolusi K-Nearest Neighbor (K-NN) yang bertujuan untuk mengurangi saat prediksi tetapi tetap mempertahankan akurasi prediksi. Metode ini masih relatif muda sehingga baru dibandingkan hanya dengan metode-metode berbasis K-NN lainnya. Dalam penelitian ini dilakukan analisis perbandingan kesamaan, perbedaan, dan kinerja terhadap metode Decision Tree (DT) dan Naïve Bayes (NB). Pengujian dengan perbandingan ini penting untuk mengetahui keunggulan dan kelemahan relatif yang dimiliki oleh K-SVNN. Dengan mengetahui keunggulan dan kelemahan maka metode tersebut dapat dibuktikan kehandalannya ketika diimplementasikan. Pengujian dilakukan baik pada saat pelatihan maupun prediksi. Kinerja pelatihan diukur dalam hal waktu yang digunakan untuk pelatihan, kinerja prediksi diukur dalam hal waktu yang digunakan untuk prediksi dan akurasi prediksi yang didapat. Hasil pengujian menunjukkan bahwa K-SVNN mempunyai akurasi yang lebih baik daripada DT dan NB. Sedangkan waktu yang digunakan untuk pelatihan dan prediksi K-SVNN lebih lama disbanding DT dan NB.

Kata kunci*:* Support Vector, Nearest Neighbor, Back-propagation, perbandingan, kinerja.

1. PENDAHULUAN

Salah satu algoritma klasifikasi yang banyak mendapat banyak perhatian para peneliti dan pembangun aplikasi adalah K-Nearest Neighbor (K-NN). K-NN juga masuk dalam jajaran 10 metode popular dalam data mining [4]. Kesederhanaan pada algoritma yang membuat K-NN mempunyai daya tarik untuk diimplementasikan dalam berbagai aplikasi. Tetapi kelemahan yang dihadapi K-NN adalah lamanya waktu yang digunakan untuk melakukan prediksi [2]. Masalah ini juga menjadi perhatian banyak peneliti untuk memperbaikinya agar kinerja waktu prediksi menjadi lebih singkat tetapi kinerja akurasi tetap baik. Karena hal tersebut maka algoritma K-NN telah mengalami banyak evolusi dalam berbagai penelitian baik untuk meningkatkan kinerja akurasi maupun kinerja waktu prediksi [5][6][7]. Hal lain yang penting untuk diketahui adalah bahwa jika K-NN versi klasik tidak ada proses pelatihan sama sekali, maka pada metode-metode berbasis K-NN yang berkembang ternyata secara implisit langung maupun tidak langsung memerlukan waktu untuk melakukan pelatihan.

Prasetyo mengusulkan kerangka kerja algoritma K-Support Vector Nearest Neighbor (K-SVNN) [1] yang bertujuan untuk melakukan reduksi pada set data latih yang digunakan untuk acuan dalam proses prediksi. Parameter penting yang berpengaruh pada hasil reduksi adalah K, dimana K adalah jumlah tetangga terdekat yang dilibatkan untuk mendapat support vector yang mempunyai pengaruh dalam penentuan keputusan hasil prediksi. Support vector yang dimaksud disini adalah data-data yang berada pada posisi disekitar garis keputusan. Garis keputusan adalah garis yang membagi data menjadi dua kelas atau lebih berbeda. Pada berbagai kasus garis ini tidak linear, tetapi K-SVNN dan metode-metode berbasis K-NN lainnya dapat memproses data dengan garis keputusan yang tidak linear. K-SVNN membutuhkan K sebagai parameter yang menentukan jumlah data yang direduksi. Pengujian kinerja metode secara lokal untuk mengamati penggunaan K menyatakan bahwa semakin kecil nilai K maka jumlah data yang tersisa sebagai support vector semakin sedikit, begitu pula sebaliknya. Hasil pengujian kinerja metode secara lokal lainnya menyatakan bahwa prediksi yang dilakukan K-SVNN tidak dipengaruhi secara signifikan oleh nilai K yang digunakan pada saat reduksi.

Pengujian kinerja secara global yang dilakukan sebelumnya adalah membandingkan K-SVNN dengan metode-metode berbasis K-NN lainnya seperti: K-NN, Template Reduction K-Nearest

Neighbor (TRKNN), dan Support Vector K-NN (SV-KNN). Hasil pengujian menyatakan bahwa kinerja akurasi prediksi dan waktu prediksi K-SVNN relatif lebih baik dibanding metode lainnya, hal ini terlihat dari akurasi yang didapatkan K-SVNN pada sebagian set data yang diujikan lebih baik daripada metode lainnya tetapi pada set data yang lain tidak lebih baik daripada metode lainnya. Sedangkan jumlah data yang dikeluarkan dari set data lebih banyak dibanding metode lainnya, tetapi untuk hal ini masih dipengaruhi nilai K yang digunakan.

Pengujian yang belum dilakukan pada K-SVNN adalah uji kinerja K-SVNN yang dibandingkan dengan metode kalsifikasi yang lain. Perbandingan kinerja yang diamati adalah waktu yang digunakan untuk pelatihan, waktu yang digunakan untuk prediksi, dan akurasi kinerja prediksi.

Makalah ini dibagi menjadi 5 bagian. Bagian 1 menyajikan pendahuluan yang melatarbelakangi penulis melakukan penelitian. Bagian 2 menyajikan penelitian-penelitian terkait yang menjadi dasar bagi penulis untuk melakukan penelitian. Bagian 3 menyajikan analisis perbandingan berbagai aspek ketiga metode yang dibandingkan. Bagian 4 menyajikan pengujian dan analisis yang dilakukan untuk mengukur kinerja ketiga metode. Dan bagian 5 menyajikan simpulan dari hasil penelitian dan saran untuk penelitian berikutnya.

2. TINJAUAN PUSTAKA

2.1 K-Support Vector Nearest Neighbor

Prasetyo [1] mengusulkan K-SVNN sebagai metode untuk mereduksi data latih sebelum melakukan prediksi. Ada waktu yang diperlukan K-SVNN untuk melakukan tahap reduksi (disebut sebagai pelatihan). Hasil reduksi adalah sejumlah data latih yang punya pengaruh pada fungsi tujuan kemudian data latih yang didapatkan tersebut disimpan untuk digunakan sebagai acuan pada saat prediksi. Prasetyo menyatakan bahwa K-SVNN termasuk dalam kategori semi eiger learning. Hasil pengujian

Parameter penting yang berpengaruh pada hasil reduksi adalah K, dimana K adalah jumlah tetangga terdekat yang dilibatkan untuk mendapat support vector yang mempunyai pengaruh dalam penentuan keputusan hasil prediksi. Support vector yang dimaksud disini adalah data-data yang berada pada posisi disekitar garis keputusan. Garis keputusan adalah garis yang membagi data menjadi dua kelas atau lebih berbeda. Pada berbagai kasus garis ini tidak linear, tetapi K-SVNN dan metode-metode berbasis K-NN lainnya dapat memproses data dengan garis keputusan yang tidak linear. K-SVNN membutuhkan K sebagai parameter yang menentukan jumlah data yang direduksi. Pengujian kinerja metode secara lokal untuk mengamati penggunaan K menyatakan bahwa semakin kecil nilai K maka jumlah data yang tersisa sebagai support vector semakin sedikit, begitu pula sebaliknya. Hasil pengujian kinerja metode secara lokal lainnya menyatakan bahwa prediksi yang dilakukan K-SVNN tidak dipengaruhi secara signifikan oleh nilai K yang digunakan pada saat reduksi. Waktu yang diperlukan untuk melakukan prediksi juga berbanding lurus terhadap nilai K yang digunakan, semakin tinggi nilai K yang digunakan maka waktu yang dibutuhkan untuk melakukan prediksi juga semakin lama, begitu pula sebaliknya. K-SVNN juga melakukan generalisasi terhadap K-NN dimana untuk K sama dengan jumlah data, maka tidak data yang dikeluarkan sehingga K-SVNN menghasilkan support vector yang sama dengan data latih sebelumnya.

Hasil uji kinerja yang dibandingkan dengan metode-metode serumpun yaitu TR-KNN dan SV-KNN menunjukkan bahwa kinerja akurasi prediksi dan waktu prediksi K-SVNN relatif lebih baik dibanding metode lainnya, hal ini terlihat dari akurasi dan waktu prediksi yang didapatkan K-SVNN pada sebagian set data yang diujikan lebih baik daripada metode lainnya tetapi pada set data yang lain tidak lebih baik daripada metode lainnya. Sedangkan jumlah data yang dikeluarkan dari set data lebih banyak dibanding metode lainnya, tetapi untuk hal ini masih dipengaruhi nilai K yang digunakan.

2.2 Decision Tree

Decision tree (pohon keputusan) adalah pohon yang digunakan sebagai prosedur penalaran untuk mendapatkan jawaban dari masalah yang dimasukkan. Pohon yang dibentuk tidak selalu berupa pohon biner. Jika semua fitur dalam data set menggunakan 2 macam nilai kategorikal maka bentuk pohon yang didapatkan berupa pohon biner. Jika dalam fitur berisi lebih dari 2 macam nilai kategorikal atau menggunakan tipe numeric maka bentuk pohon yang didapatkan biasanya tidak berupa pohon biner [2].

Kefleksibelan membuat metode ini atraktif, khususnya karena memberikan keuntungan berupa visualisasi saran (dalam bentuk pohon keputusan) yang membuat prosedur prediksinya dapat diamati. Decision tree banyak digunakan untuk menyelesaikan kasus penentuan keputusan seperti

dibidang kedokteran (diagnosis penyakit pasien), ilmu komputer (struktur data), psikologi (teori pengambilan keputusan), dan sebagainya.

Karakteristik dari decision tree dibentuk dari sejumlah elemen sebagai berikut [2]:

1. Node akar, tidak mempunyai lengan masukan dan mempunyai nol atau lebih lengan keluaran
2. Node internal, setiap node yang bukan daun (non-terminal) yang mempunyai tepat satu lengan masukan dan dua atau lebih lengan keluaran. Node ini menyatakan pengujian yang didasarkan pada nilai fitur.
3. Lengan, setiap cabang menyatakan nilai hasil pengujian di node bukan daun
4. Node daun (terminal), node yang mempunyai tepat satu lengan masukan dan tidak mempunyai lengan keluaran. Node ini menyatakan label kelas (keputusan)

Decision tree mempunyai tiga pendekatan klasik:

1. Pohon klasifikasi, digunakan untuk melakukan prediksi ketika ada data baru yang belum diletahui label kelasnya. Pendekatan ini yang paling banyak digunakan.
2. Pohon regresi, ketika hasil prediksi dianggap sebagai nilai nyata yang mungkin akan didapatkan. Misalnya, kasus hanya minyak, kenaikan harga rumah, prediksi inflasi tiap tahun, dan sebagainya.
3. CART (atau C&RT), ketika masalah klasifikasi dan regresi digunakan bersama-sama.

Jika memperhatikan kriteria pemilihan cabang pemecah, maka algoritma penginduksi pohon keputusan mempunyai beberapa macam:

1. GINI (*impurity*) index
2. entropy (*impurity*)
3. misclassification
4. Chi-square
5. G-square

2.3 Naïve Bayes

Bayes merupakan teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema Bayes (atau aturan Bayes) dengan asumsi independensi (ketidaktergantungan) yang kuat (naif). Dengan kata lain, dalam Naïve Bayes, model yang digunakan adalah "model fitur independen" [3].

Untuk mengaitkan Naïve Bayes dengan klasifikasi, korelasi hipotesis dan evidence dengan klasifikasi adalah bahwa hipotesis dalam teorema Bayes merupakan label kelas yang menjadi target pemetaan dalam klasifikasi, sedangkan evidence merupakan fitur-fitur yang menjadi masukan dalam model klasifikasi. Jika X adalah vektor masukan yang berisi fitur, dan Y adalah label kelas, maka Naïve Bayes dituliskan dengan P(Y|X), notasi ini berarti probabilitas label kelas Y didapatkan setelah fitur-fitur X diamati, notasi ini disebut juga probabilitas akhir (*posterior probability*) untuk Y. Sedangkan P(Y) disebut probabilitas awal (*prior probability*) Y.

Selama proses pelatihan, harus dilakukan pembelajaran probabilitas akhir (P(Y|X) pada model untuk setiap kombinasi X dan Y berdasarkan informasi yang didapat dari data latih. Dengan membangun model tersebut, maka untuk suatu data uji X' dapat diklasifikasikan dengan mencari nilai Y' dengan memaksimalkan nilai P(Y'|X') yang didapat [3].

3. ANALISIS PERBANDINGAN

Pada penelition ini, metode-metode yang dilakukan pengujian kinerja dan analisis yaitu K-SVNN, Decision Tree (DT), dan Naive Bayes (NB). Ketiga metode ini dapat digunakan untuk klasifikasi, tetapi berasal dari rumpun yang berbeda. K-SVNN diturunkan dari K-NN. DT merupakan teknik klasifikasi menggunakan struktur pohon keputusan dalam melakukan prediksi, pada penelitian ini metode induksi pohon keputusannya adalah C4.5. Sedangkan NB merupakan teknik klasifikasi yang mengadopsi konsep probabilistik. Karena berasal dari rumpun yang berbeda maka penggunaan parameter-parameter dalam penggunaannya juga berbeda, tetapi ketiga metode bertujuan sama, yaitu melakukan klasifikasi. K-SVNN yang dibandingkan terhadap DT dan NB tidak dibandingkan dalam hal parameter melainkan dalam hal kinerja, baik kinerja pada saat pelatihan maupun pada saat prediksi. Parameter-parameter untuk ketiga metode dipilih nilai-nilai yang dapat mengoptimalkan akurasi. Kinerja pelatihan diukur dalam hal waktu yang digunakan untuk pelatihan, kinerja prediksi diukur dalam hal waktu yang digunakan untuk prediksi dan akurasi prediksi yang didapat.

Hasil analisis yang dilakukan penulis dalam menemukan persamaan yang dimiliki oleh ketiga metode tersebut adalah sebagai berikut:

1) Ketiga metode memerlukan proses pelatihan sebelum model digunakan pada saat prediksi.
2) Ketiga metode dapat memproses data-data yang mempunyai garis keputusan yang tidak linear.

Tabel 1. Perbedaan metode K-SVNN, DT, dan NB

Kriteria	K-SVNN	DT	NB
Penyimpanan sebagian data latih	Ya	Tidak	Tidak
Kriteria yang mempengaruhi kinerja	K tetangga terdekat	Kriteria pemilihan fitur sebagai cabang	-
Solusi global optima	Tidak	Tidak	Ya
Kebutuhan memori	Besar	Kecil	Kecil

Sedangkan perbedaan ketiga metode disajikan pada tabel 1. Hasil analisis pada saat pengamatan proses metode dapat dijelaskan sebagai berikut:

1) Penyimpanan sebagian set data latih
   DT dan NB sama sekali tidak menyimpan satupun data yang digunakan pada saat pelatihan. DT hanya menyimpan variabel struktur pohon yang didapat pada saat pelatihan saja yang disimpan. NB hanya menyimpan nilai-nilai probabilitas hasil perhitungan setiap variasi nilai dari semua fitur. Sedangkan K-SVNN menyimpan sebagian data yang berpengaruh pada fungsi tujuan.
2) Kriteria yang mempengaruhi kinerja
   DT hanya menggunakan parameter jenis kriteria untuk pemilihan fitur yang mencadi cabang dari pohon yang dibentuk, seperti Gini, dan Gain. Tidak ada pilihan nilai untuk penggunakan Gini atau Gain. Sedangkan NB tidak menggunakan parameter apapun dalam proses pelatihan. Sedangkan K-SVNN menggunakan K tetangga terdekat. Pemilihan nilai K juga menjadi hal yang sensitif.
3) Solusi global optima
   Solusi gobal optima merupakan solusi yang selalu mengarah pada jawaban yang sama pada setiap kali percobaan. Hanya DT yang bisa dipastikan mengarah pada solusi yang global optima. Sedangkan K-SVNN relatif dipengaruhi oleh nilai K yang digunakan, untuk K yang sama pada setiap percobaan K-SVNN dapat mengarah pada solusi global optima, tetapi untuk K berbeda pada setiap percobaan K-SVNN dapat terjebak pada solusi lokal optima.
4) Kebutuhan memori
   DT hanya membutuhkan sejumlah variabel untuk menyimpan variabel struktur pohon yang jumlahnya relatif sedikit. NB juga membutuhkan sejumlah variabel untuk menyimpan nilai-nilai probabilitas hasil perhitungan setiap variasi nilai dari semua fitur. Keduanya membutuhkan memori yang relatif sedikit. Sedangkan ukuran memori yang dibutuhkan K-SVNN sejumlah N kuadrat untuk menyimpan jarak antara pasangan dua data, dimana N adalah jumlah data, dan ini sangat besar.

4. PENGUJIAN KINERJA DAN ANALISIS HASIL

Pengujian dilakukan terhadap empat set data publik yang diunduh dari UCI Machine Learning Repository [8], yaitu: Iris (150 record, 4 fitur), Vertebral Column (310 record, 6 fitur), Wine (178 record, 13 fitur), dan Glass (214 record, 9 fitur). Sistem pengujian menggunakan 5 fold, dimana 80% digunakan sebagai data latih dan 20% digunakan sebagai data uji. K-SVNN yang diuji dalam penelitian ini masih bekerja hanya pada dua kelas saja, sehingga harus dilakukan penggabungan beberapa kelas berbeda menjadi satu kelas pada data set yang komposisi kelasnya lebih dari dua, yaitu Iris, dilakukan pengabungan data dengan label kelas ‘setosa’ dan ‘versicolor’ menjadi satu kelas. Karena data-data pada setiap fitur mempunyai jangkauan nilai yang berbeda,maka dilakukan pra-pemrosesan yaitu normalisasi. Sebelum dilakukan proses pengujian, semua data pada setiap fitur dilakukan normalisasi agar nilai pada setiap fitur menggunakan jangkauan yang sama yaitu [0,1]. Untuk K-SVNN, pengujian dilakukan menggunakan nilai K = 13 baik untuk pelatihan maupun prediksi.

Hasil pengujian untuk akurasi disajikan pada Tabel 2, hasil pengujian untuk waktu yang digunakan dalam proses pelatihan disajikan pada Tabel 3, hasil pengujian untuk waktu yang digunakan dalam proses prediksi disajikan pada tabel 4.

Dari hasil yang disajikan pada Tabel 2, dapat diamati bahwa K-SVNN mempunyai akurasi prediksi yang lebih baik daripada metode pembanding, kolom keterangan memberikan point keunggulan K-SVNN dibanding metode lainnya. K-SVNN mempunyai akurasi kinerja lebih baik minimal 18%, seperti pada set data Glass antara K-SVNN dibandingkan DT.

Hasil pengujian waktu yang digunakan selama proses pelatihan menunjukkan bahwa K-SVNN

lebih lama dibanding metode pembanding yaitu DT dan NB. Selisih yang didapatkan, paling sedikit 10 milidetik dan paling banyak 58 milidetik.

Tabel 2. Akurasi prediksi

Set data	Akurasi (%)			Selisih terkecil	Ket.
	K-SVNN	DT	NB		
Iris	94.00	63.00	46.00	31.00%	**
Ver. Col.	79.03	18.39	28.06	50.97%	**
Wine	84.27	56.19	59.62	24.65%	**
Glass	89.71	71.51	71.48	18.20%	**

Keterangan: (*) berarti K-SVNN unggul dari 1 metode yang lain, (**) berarti K-SVNN unggul dari 2 metode yang lain.

Tabel 3. Waktu pelatihan

Set data	Waktu (milidetik)			Ket.
	K-SVNN	DT	NB	
Iris	20.92	10.55	3.29	
Ver. Col.	61.98	23.52	3.58	
Wine	35.25	10.62	10.46	
Glass	44.52	12.59	3.62	

Tabel 4. Waktu prediksi

Set data	Waktu (milidetik)			Ket.
	K-SVNN	DT	NB	
Iris	1.76	0.32	1.87	*
Ver. Col.	5.93	0.37	2.01	
Wine	3.02	0.34	1.96	
Glass	6.98	0.33	2.02	

Keterangan: (*) berarti K-SVNN unggul dari 1 metode yang lain, (**) berarti K-SVNN unggul dari 2 metode yang lain.

Hasil pengujian untuk waktu prediksi menunjukkan bahwa K-SVNN masih kalah dibandingkan kedua metode pembanding pada semua set data, dengan selisih paling kecil 0.1 milidetik. Hal ini sangat beralasan karena DT dan NB tidak menggunakan sama sekali set latih yang sudah dilatihkan terhadapnya sehingga proses prediksi menjadi lebih singkat.

Dari analisis pengujian yang dilakukan pada 3 masalah tersebut, dapat dinyatakan bahwa K-SVNN hanya unggul pada satu sisi yaitu akurasi, sedangkan waktu yang digunakan untuk pelatihan prediksi lebih lama dibanding DT dan NB.

5. SIMPULAN

Dari pengujian dan analisis yang dilakukan dalam penelitian ini dapat disimpulkan sebagai berikut:

1) K-SVNN mempunyai akurasi yang lebih baik daripada DT dan NB dengan selisih lebih baik paling kecil 18.20%.
2) K-SVNN mempunyai waktu pelatihan dan prediksi yang lebih lama dibanding DT dan NB.

Saran yang dapat diberikan dari hasil penelitian ini adalah sebagai berikut:

1) Pengujian dalam penelitian ini hanya diterapkan pada 4 set data saja, sehingga hasil yang didapat dari penelitian ini masih relatif terhadap set data yang sudah diuji saja. Perlu pendalaman lebih lanjut dengan mengujinya pada set data yang lain.
2) K-SVNN masih perlu dibandingkan dengan metode-metode klasifikasi yang lain, seperti: boosting atau hidden markov, untuk mengukur sejauh mana kinerja metode-metode tersebut ketika diimplementasikan.

DAFTAR PUSTAKA

[1] Prasetyo, E., 2012. K-Support Vector Nearest Neighbor untuk Klasifikasi Berbasis K-NN, in proceding *Seminar Nasional Sistem Informasi Indonesia*, Jurusan Sistem Informati ITS, Surabaya.

[2] Tan, P., Steinbach, M., Kumar, V., 2006. *Introduction to Data Mining*, 1st Ed, Pearson Education: Boston San Fransisco New York.

[3] Prasetyo, E., 2012. *Data Mining – Konsep dan Aplikasi Menggunakan Matlab*, edisi 1, Andi Offset: Yogyakarta.

[4] Wu, X., Kumar, V., 2009. *The Top Ten Algorithms in Data Mining*, CRC Press Taylor & Francis Group: Boca Raton London.

[5] Gowda, K.C., Krishna, G. 1979. *The Condensed Nearest Neighbor Rule Using the Concept of Mutual Nearest Neighborhood*. IEEE Transactions on Information Theory. 25 (4), pp.488-490.

[6] Srisawat, A., Phienthrakul, T., Kijsirikul, B. 2006. SV-KNNC: An Algorithm for Improving the Efficiency of K-Nearest Neighbor. In: Qiang Yang, Geoffrey I. Webb. *The 09th Pacific Rim International Conference on Artificial Intelligence* (PRICAI-2006). Guilin, China, 7-11 August 2006. Springer-Verlag Berlin Heidelberg.

[7] Fayed, H.A., Atiya, A.F. 2009. *A Novel Template Reduction Approach for the K-Nearest Neghbor Method*. IEEE Transaction on Neural Network, 20(5), pp.890-896.

[8] *UCI Machine Learning Repository* , 20 Mei 2012, http://archive.ics.uci.edu/ml/datasets.html